Parametric and Quantitative Extensions of Modal Transition Systems

Uli Fahrenberg, Kim Guldstrand Larsen, Axel Legay, Louis-Marie Traonouez

Parametric and Quantitative Extensions of Modal Transition Systems

Uli Fahrenberg[1], Kim G. Larsen[2], Axel Legay[1], and Louis-Marie Traonouez[1]

[1] Inria / IRISA, Rennes, France
[2] Aalborg University, Aalborg, Denmark

Abstract. Modal transition systems provide a behavioral and compositional specification formalism for reactive systems. We survey two extensions of modal transition systems: parametric modal transition systems for specifications with parameters, and weighted modal transition systems for quantitative specifications.

1 Introduction

Modal transition systems [21, 23] provide a behavioral and compositional specification formalism for reactive systems. They grew out of the notion of relativized bisimulation [20], which allows for simple specifications of components by allowing the notion of bisimulation to take into account the restricted use that a given component may have in its context.

A modal transition system is essentially a (labeled) transition system, but with two types of transitions: so-called *may*-transitions which any implementation may (or may not) have, and *must*-transitions which any implementation is required to have. In fact, ordinary labeled transition systems (or implementations) are modal transition systems where the set of may- and must-transitions coincide. Modal transition systems come equipped with a bisimulation-like notion of (modal) refinement, reflecting that the more must-transitions and the fewer may-transitions a modal specification has the more refined and closer to a final implementation it is.

Example 1. Consider the modal transition system shown in Fig. 1 which models the requirements of a simple email system in which emails are first received and then delivered; must- and may-transitions are represented by solid and dashed arrows, respectively. Before delivering the email, the system may check or process the email, *e.g.* for encryption or decryption, filtering of spam emails, or generating automatic answers using an auto-reply feature. Any implementation of this email system specification *must* be able to receive and deliver email, and it *may* also be able to check arriving email before delivering it. No other behavior is allowed. Such a valid implementation is given in Fig. 2.

The theory of modal transition systems (MTS), or *modal specifications* as they were called in the paper [21] in the proceedings of the first CAV conference

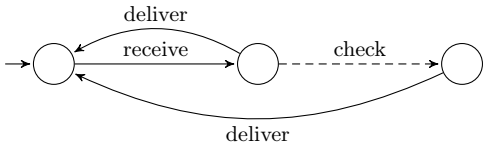


Fig. 1: Modal transition system modeling a simple email system, with an optional behavior: Once an email is received, it may be checked, *e.g.* be scanned for containing viruses, or automatically decrypted, before it is delivered to the receiver.

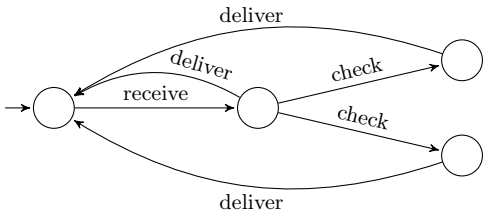


Fig. 2: An implementation of the simple email system in Fig. 1 in which we explicitly model two distinct types of email pre-processing.

organized by Joseph Sifakis in Grenoble,[3] was aiming at providing a *behavioral* compositional specification formalism for reactive systems. At the time of the introduction of MTS, there were two predominant approaches to specifications formalisms and verification methods for reactive and concurrent systems: *logical* approaches where a specification is a set of properties of implementations (labeled transition systems), and *graphical* approaches promoted by the various process algebras, where implementations and specifications are systems of the same kind – namely labeled transition systems, and verification amounts to compare such systems with respect to a given behavioral preorder, *e.g.* bisimilarity.

In search for a complete specification theory, the following properties have been considered desirable (the first three were listed in the early paper [6]):

expressiveness: the specification formalism should be powerful enough to express all properties of a given implementation. In other words it should be possible to completely specify any labeled transition system, up to bisimulation.

modularity: implementations are often made out of several components, and it should be possible to infer satisfaction of an overall specification solely on the basis of sub-specification of the sub-components.

refinement: one should have the ability to deal with partial specifications, requiring more and more properties about a system, up to its complete specification.

[3] In fact, the first CAV conference was not called CAV, but had the rather lengthy title "Automatic Verification Methods for Finite State Systems."

logical composition: specification should be composable with respect to usual logical operators such as conjunction and (possibly) disjunction.

quotienting: given an overall specification S of a composite systems as well as a sub-specification T of a sub-component, the existence of a quotient specification $S\backslash T$ will describe the sufficient and necessary condition of the remaining components in order that S is satisfied by the total systems.

Applying these criteria to the logical and graphical (*i.e.* bisimulation) framework, as was done in [6], we see that the logical and graphical frameworks offer complementary advantages: on the graphical side, expressiveness is trivial since a process i a specification of itself. Modularity is usually guaranteed by the fact that bisimulations are compatible with (most) process constructors. On the logical side, expressiveness is achieved if we allow possibly infinite sets of formulae as logical specifications, or admit recursively specified properties. The point of modularity has proved more difficult with early attempts of Sifakis and Graf [15] and Holmstrøm [17] providing sound and highly usable proof systems for specifications mixing logical and behavioral constructs (as well as fix-point constructs) but lacking accompanying completeness results. Much later the work of Mardare and Policriti [25] provided a first matching completeness result.

In the rest of this paper, we survey two extensions of modal transition systems. The first extension, *parametric* modal transition systems, is concerned with systems whose behaviors depend on parameters [4]. The second extension, *weighted* modal transition systems [1, 2] permits to reason on systems whose behaviors depend on quantities. Another paper in this volume [11] will be concerned with other extensions of modal transition systems which are more closely related to applications.

Acknowledgment. This survey paper presents research which we have conducted with a number of coauthors; in alphabetical order, these are Sebastian S. Bauer, Nikola Beneš, Line Juhl, Jan Křetínský, Mikael H. Møller, Jiří Srba, and Claus Thrane. We acknowledge their cooperation in this work; any errors in this presentation are, however, our own.

2 Parametric Modal Transition Systems

It is well admitted (see *e.g.* [27]) that MTS and their extensions like disjunctive MTS (DMTS) [24], 1-selecting MTS (1MTS) [13] and transition systems with obligations (OTS) [5] provide strong support for a specification formalism allowing for step-wise refinement process. Moreover, the MTS formalisms have applications in other contexts, which include verification of product lines [16, 22], interface theories [27, 28] and modal abstractions in program analysis [14, 18, 26].

Unfortunately, all of these formalisms lack the capability to express some intuitive specification requirements like exclusive, conditional and persistent choices. In [4] the expressive power of MTS and its variants has been extended considerably so it can model model arbitrary Boolean conditions on transitions

and also allows to instantiate persistent transitions. The model, called *parametric modal transition systems* (PMTS), is equipped with a finite set of parameters that are fixed prior to the instantiation of the transitions in the specification. The generalized notion of modal refinement is designed to handle the parametric extension and it specializes to the well-studied modal refinements on all the subclasses of our model like MTS, disjunctive MTS and MTS with obligations.

2.1 Motivation

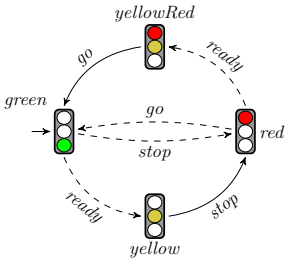
We shall now discuss these limitations on an example as a motivation for the introduction of parametric MTS formalism with general Boolean conditions in specification requirements.

Consider a simple specification of a traffic light controller that can be at any moment in one of the four predefined states: *red*, *green*, *yellow* or *yellowRed*. The requirements of the specification are: when *green* is on the traffic light may either change to *red* or *yellow* and if it turned *yellow* it must go to *red* afterward; when *red* is on it may either turn to *green* or *yellowRed*, and if it turns *yellowRed* (as it is the case in some countries) it must go to *green* afterwords.

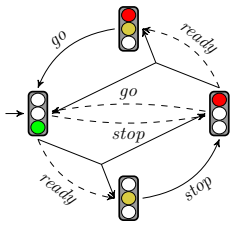
Fig. 3a shows an obvious MTS specification of the proposed specification. The transitions in the standard MTS formalism are either of type may (optional transitions depicted as dashed lines) or must (required transitions depicted as solid lines). In Fig. 3c, Fig. 3d and Fig. 3e we present three different implementations of the MTS specification where there are no more optional transitions. The implementation I_1 does not implement any may transition as it is a valid possibility to satisfy the specification S_1. Of course, in our concrete example, this means that the light is constantly *green* and it is clearly an undesirable behavior that cannot be, however, easily avoided. The second implementation I_2 on the other hand implements all may transitions, again a legal implementation in the MTS methodology but not a desirable implementation of a traffic light as the next action is not always deterministically given. Finally, the implementation I_3 of S_1 illustrates the third problem with the MTS specifications, namely that the choices made in each turn are not persistent and the implementation alternates between entering *yellow* or not. None of these problems can be avoided when using the MTS formalism.

A more expressive formalism of disjunctive modal transition systems (DMTS) can overcome some of the above mentioned problems. A possible DMTS specification S_2 is depicted in Fig. 3b. Here the *ready* and *stop* transitions, as well as *ready* and *go* ones, are disjunctive, meaning that it is still optional which one is implemented but at least one of them must be present. Now the system I_1 in Fig. 3c is not a valid implementation of S_2 any more. Nevertheless, the undesirable implementations I_2 and I_3 are still possible and the modeling power of DMTS is insufficient to eliminate them.

Inspired by the recent notion of transition systems with obligations [5], we can model the traffic light using specification as a transition system with

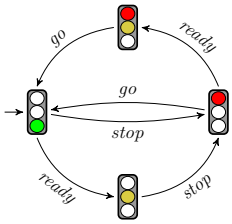


(a) MTS specification S_1

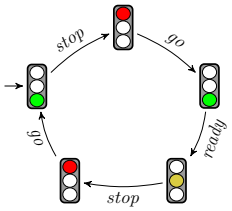


(b) DMTS specification S_2

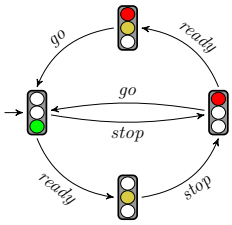
(c) Implementation I_1



(d) Implementation I_2

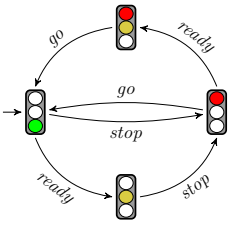


(e) Implementation I_3



Obligation function:
$\Phi(green) = (stop, red) \oplus (ready, yellow)$
$\Phi(red) = (go, green) \oplus (ready, yellowRed)$

(f) Specification S_3



Parameters: $\{reqYfromR, reqYfromG\}$
Obligation function:
$\Phi(green) = ((stop, red) \oplus (ready, yellow))$
$\wedge(reqYfromG \Leftrightarrow (ready, yellow))$
$\Phi(red) = ((go, green) \oplus (ready, yellowRed))$
$\wedge(reqYfromR \Leftrightarrow (ready, yellowRed))$

(g) PMTS specification S_4

Fig. 3: Specifications and implementations of a traffic light controller

arbitrary[4] obligation formulae. These formulae are Boolean propositions over the outgoing transitions from each state, whose satisfying assignments yield the allowed combinations of outgoing transitions. A possible specification called S_3 is given in Fig. 3f and it uses the operation of exclusive-or. We will follow an agreement that whenever the obligation function for some node is not listed in the system description then it is implicitly understood as requiring all the available outgoing transitions to be present. Due to the use of exclusive-or in the obligation function, the transition systems I_1 and I_2 are not valid implementation

[4] In the transition systems with obligations only positive Boolean formulae are allowed.

any more. Nevertheless, the implementation I_3 in Fig. 3e cannot be avoided in this formalism either.

Finally, the problem with the alternating implementation I_3 is that we cannot enforce in any of the above mentioned formalisms a uniform (persistent) implementation of the same transitions in all its states. In order to overcome this problem, we propose the so-called parametric MTS where we can, moreover, choose persistently whether the transition to *yellow* is present or not via the use of parameters. The PMTS specification with two parameters *reqYfromR* and *reqYfromG* is shown in Fig. 3g. Fixing a priori the (Boolean) values of the parameters makes the choices permanent in the whole implementation, hence we eliminate also the last problematic implementation I_3.

2.2 Definition

We shall now formally capture the intuition behind parametric MTS introduced above. First, we recall the standard propositional logic.

A Boolean formula over a set X of atomic propositions is given by the following abstract syntax

$$\varphi ::= \mathbf{tt} \mid x \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

where x ranges over X. The set of all Boolean formulae over the set X is denoted by $\mathcal{B}(X)$. Let $\nu \subseteq X$ be a truth assignment, *i.e.* a set of variables with value true, then the satisfaction relation $\nu \models \varphi$ is given by $\nu \models \mathbf{tt}$, $\nu \models x$ iff $x \in \nu$, and the satisfaction of the remaining Boolean connectives is defined in the standard way. We also use the standard derived operators like exclusive-or $\varphi \oplus \psi = (\varphi \wedge \neg\psi) \vee (\neg\varphi \wedge \psi)$, implication $\varphi \Rightarrow \psi = \neg\varphi \vee \psi$ and equivalence $\varphi \Leftrightarrow \psi = (\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi)$.

We can now proceed with the definition of parametric MTS.

Definition 1. *A* parametric MTS *(PMTS) over an action alphabet Σ is a tuple (S, T, P, Φ) where S is a set of* states*, $T \subseteq S \times \Sigma \times S$ is a* transition relation*, P is a finite set of* parameters*, and $\Phi : S \to \mathcal{B}((\Sigma \times S) \cup P)$ is an* obligation function *over the atomic propositions containing outgoing transitions and parameters. We implicitly assume that whenever $(a, t) \in \Phi(s)$ then $(s, a, t) \in T$. By $T(s) = \{(a, t) \mid (s, a, t) \in T\}$ we denote the set of all outgoing transitions of s.*

PMTS has been provided a refinement notion that generalizes the well-studied refinement notions on its subclasses including that of MTS. In the definition, the parameters are fixed first (persistence) followed by all valid choices modulo the fixed parameters that now behave as constants.

First we set the following notation. Let (S, T, P, Φ) be a PMTS and $\nu \subseteq P$ be a truth assignment. For $s \in S$, we denote by $\mathrm{Tran}_\nu(s) = \{E \subseteq T(s) \mid E \cup \nu \models \Phi(s)\}$ the set of all admissible sets of transitions from s under the fixed truth values of the parameters.

We can now define the notion of modal refinement between PMTS.

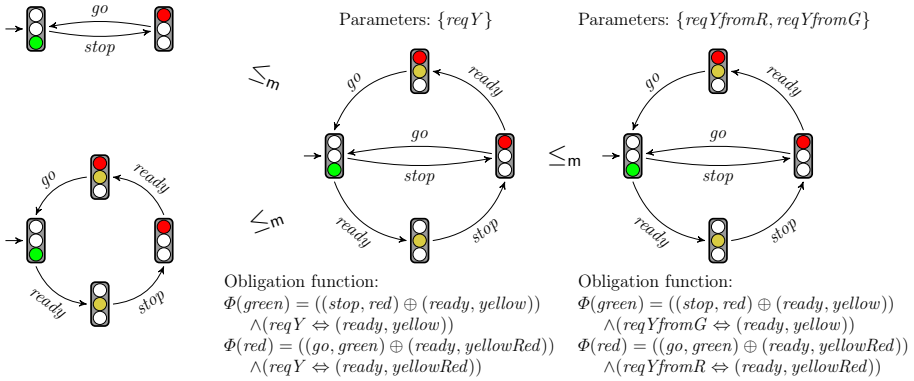


Fig. 4: Example of modal refinement

Definition 2. *Let* (S_1, T_1, P_1, Φ_1) *and* (S_2, T_2, P_2, Φ_2) *be two PMTS. A binary relation* $R \subseteq S_1 \times S_2$ *is a* modal refinement *if for each* $\mu \subseteq P_1$ *there exists* $\nu \subseteq P_2$ *such that for every* $(s,t) \in R$ *holds*

$$\forall M \in \mathit{Tran}_\mu(s) : \exists N \in \mathit{Tran}_\nu(t) : \ \forall (a,s') \in M : \exists (a,t') \in N : (s',t') \in R \ \wedge \\ \forall (a,t') \in N : \exists (a,s') \in M : (s',t') \in R \ .$$

We say that s *modally refines* t*, denoted by* $s \leq_{\mathsf{m}} t$*, if there exists a modal refinement* R *such that* $(s,t) \in R$*.*

Example 2. Consider the rightmost PMTS in Fig. 4. It has two parameters *reqYfromG* and *reqYfromR* whose values can be set independently and it can be refined by the system in the middle of the figure having only one parameter *reqY*. This single parameter simply binds the two original parameters to the same value. The PMTS in the middle can be further refined into the implementations where either *yellow* is always used in both cases, or never at all. Notice that there are in principle infinitely many implementations of the system in the middle, however, they are all bisimilar to either of the two implementations depicted in the left of Fig. 4.

[4] provides an extensive study of the complexity of refinement checking between parametric modal transitions with classification depending on the complexity of obligations as well as the presence or absence of parameters. For each combination the complexity class of the polynomial hierarchy for which modal refinement is complete is provided. In short, the complexities ranges from P-complete to Π_4^p-complete (thus in PSPACE).

3 Quantitative Modal Transition Systems

Motivated by applications to embedded, real-time and hybrid systems, the modal transition system framework has been extended in order to reason about

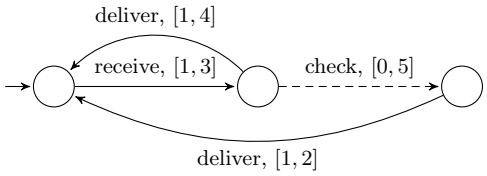


Fig. 5: Specification of a simple email system, similar to Fig. 1, but extended by integer intervals modeling time units for performing the corresponding actions.

quantitative aspects [3, 19]. With these applications in mind, it is necessary not only to be able to *specify* quantitative aspects of systems, but also to formalize successive *refinement* of quantities. To illustrate this extension, consider again the modal transition system of Fig. 1, but this time with quantities, see Fig. 5: Every transition label is extended by integer intervals modeling upper and lower bounds on time required for performing the corresponding actions. For instance, the reception of a new email (action *receive*) must take between one and three time units, the checking of the email (action *check*) is allowed to take up to five time units.

In this quantitative setting, there is a problem with using a *Boolean* notion of refinement as is done in the preceding section: If one only can decide *whether or not* an implementation refines a specification, then the quantitative aspects get lost in the refinement process. As an example, consider the email system implementations in Fig. 6. Implementation (a) does not refine the specification, as there is an error in the discrete structure of actions: after receiving an email, the system can check it indefinitely without ever delivering it. Also implementations (b) and (c) do not refine the specification: (b) takes too long to receive email, (c) does not deliver email fast enough after checking it. Implementation (d) on the other hand is a perfect refinement of the specification.

Intuitively however, implementations (b) and (c) conform much better to the specification than implementation (a) in Fig. 6: there are no discrepancies in the discrete structure, only the weights are off by 1. Additionally, the quantitative error in implementation (c) occurs later than the one in (b). Hence one may want to say that implementation (d) is in perfect refinement of the specification, (c) is slightly off, (b) is a bit more problematic, whereas implementation (a) is completely unacceptable. A Boolean notion of refinement does not allow to make such distinctions between different negative answers.

To sum up, a Boolean notion of refinement is too *fragile* for quantitative formalisms. Minor and major modifications in the implementation cannot be distinguished, as both of them may reverse the Boolean answer. As observed *e.g.* in [9], this view is obsolete; engineers need quantitative notions on how modified implementations differ. The introduction of such a quantitative notion of refinement, and its consequences for the specification theory, are the subject of this section, which is based on the papers [1, 2].

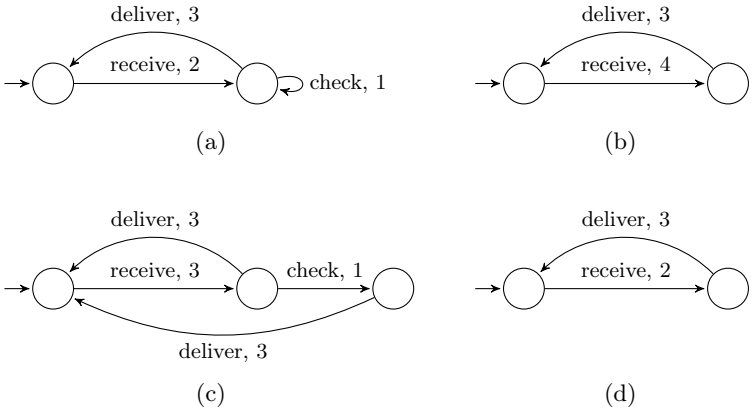


Fig. 6: Four implementations of the simple email system in Fig. 5.

Depending on the precise application of our quantitative formalism, there are a few choices which one has to make. One such choice is the precise definition of quantitative refinement, as the way quantitative discrepancies between specifications is measured *e.g.* depends on whether differences accumulate over time or the interest more lies in the maximal individual differences. Another choice is how to combine quantities during structural composition: when modeling *e.g.* energy consumption, they should be added; when modeling timing constraints, some form of conjunction should be used.

To facilitate quantitative reasoning on specifications and implementations, we introduce a real-valued *distance* between specifications such that perfect refinement corresponds to distance 0, small quantitative discrepancies give rise to small distances, and differences in the discrete control structure correspond to distance ∞. For the examples in Figs. 5 and 6, we will deduce the following chain of decreasing distances:

$$\infty = d(I_1, S) > d(I_2, S) > d(I_3, S) > d(I_4, S) = 0$$

3.1 Weighted modal transition systems

Let Σ be a set of labels with a preorder $\sqsubseteq \subseteq \Sigma \times \Sigma$, and denote by $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$ the set of finite and infinite traces over Σ. $\mathsf{len}(\sigma)$, for $\sigma \in \Sigma^\infty$, denotes the length (finite or infinite) of a trace σ. Let $\varepsilon \in \Sigma^\infty$ denote the empty trace, and for $a \in \Sigma$, $\sigma \in \Sigma^\infty$, denote by $a.\sigma$ their concatenation.

A *weighted modal transition system* (WMTS) is a tuple $\mathcal{S} = (S, s^0, \dashrightarrow, \longrightarrow)$ consisting of a set S of states, an initial state $s_0 \in S$, and must- and may-transitions $\longrightarrow, \dashrightarrow \subseteq S \times \Sigma \times S$ for which it holds that for all $s \xrightarrow{a} s'$ there is $s \overset{b}{\dashrightarrow} s'$ with $a \sqsubseteq b$.

Intuitively, a may-transition $s \overset{b}{\dashrightarrow} t$ specifies that an implementation $\mathcal{I}$ of $\mathcal{S}$ is *permitted* to have a corresponding transition $i \overset{a}{\longrightarrow} j$, for any $a \sqsubseteq b$, whereas a must-transition $s \overset{b}{\longrightarrow} t$ postulates that $\mathcal{I}$ is *required* to implement at least one corresponding transition $i \overset{a}{\longrightarrow} j$ for some $a \sqsubseteq b$. We will make this precise below.

An WMTS $\mathcal{S} = (S, s^0, \dashrightarrow, \longrightarrow)$ is an *implementation* if $\longrightarrow = \dashrightarrow$. Hence in an implementation, all optional behavior has been resolved.

Definition 3. *A* modal refinement *of WMTS* $\mathcal{S}_1 = (S_1, s_1^0, \dashrightarrow_1, \longrightarrow_1)$, $\mathcal{S}_2 = (S_2, s_2^0, \dashrightarrow_2, \longrightarrow_2)$ *is a relation* $R \subseteq S_1 \times S_2$ *such that for any* $(s_1, s_2) \in R$,

- *whenever* $s_1 \overset{a_1}{\dashrightarrow}_1 t_1$, *then also* $s_2 \overset{a_2}{\dashrightarrow}_2 t_2$ *for some* $a_1 \sqsubseteq a_2$ *and* $(t_1, t_2) \in R$,
- *whenever* $s_2 \overset{a_2}{\longrightarrow}_2 t_2$, *then also* $s_1 \overset{a_1}{\longrightarrow}_1 t_1$ *for some* $a_1 \sqsubseteq a_2$ *and* $(t_1, t_2) \in R$.

Thus any behavior which is permitted in $\mathcal{S}_1$ is also permitted in $\mathcal{S}_2$, and any behavior required in $\mathcal{S}_2$ is also required in $\mathcal{S}_1$. We write $\mathcal{S}_1 \leq_{\mathsf{m}} \mathcal{S}_2$ if there is a modal refinement $R \subseteq S_1 \times S_2$ with $(s_1^0, s_2^0) \in R$.

The *implementation semantics* of a WMTS $\mathcal{S}$ is the set $[\![\mathcal{S}]\!] = \{\mathcal{I} \leq_{\mathsf{m}} \mathcal{S} \mid \mathcal{I} \text{ implementation}\}$, and we write $\mathcal{S}_1 \leq_{\mathsf{t}} \mathcal{S}_1$ if $[\![\mathcal{S}_1]\!] \subseteq [\![\mathcal{S}_2]\!]$, saying that $\mathcal{S}_1$ *thoroughly refines* $\mathcal{S}_2$. It follows by transitivity of $\leq_{\mathsf{m}}$ that $\mathcal{S}_1 \leq_{\mathsf{m}} \mathcal{S}_2$ implies $\mathcal{S}_1 \leq_{\mathsf{t}} \mathcal{S}_2$, hence modal refinement is a *syntactic over-approximation* of thorough refinement.

3.2 Distances

Recall that a *hemimetric* on a set X is a function $d : X \times X \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ which satisfies $d(x, x) = 0$ and $d(x, y) + d(y, z) \geq d(x, z)$ (the *triangle inequality*) for all $x, y, z \in X$. Note that our hemimetrics are *extended* in that they can take the value ∞.

We will need to generalize hemimetrics to codomains other than $\mathbb{R}_{\geq 0} \cup \{\infty\}$. For a partially ordered monoid $(\mathbb{L}, \sqsubseteq, \oplus, \mathbb{0})$, an $\mathbb{L}$-*hemimetric* on X is a function $d : X \times X \to \mathbb{L}$ which satisfies $d(x, x) = \mathbb{0}$ and $d(x, y) \oplus d(y, z) \sqsupseteq d(x, z)$ for all $x, y, z \in X$.

Definition 4. *A* trace distance *is a hemimetric* $td : \Sigma^\infty \times \Sigma^\infty \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ *for which* $td(a, b) = 0$ *for all* $a, b \in \Sigma$ *with* $a \sqsubseteq b$ *and* $td(\sigma, \tau) = \infty$ *whenever* $\mathsf{len}(\sigma) \neq \mathsf{len}(\tau)$.

For any set M, let $\mathbb{L}M = (\mathbb{R}_{\geq 0} \cup \{\infty\})^M$ the set of functions from M to the extended non-negative real line. Then $\mathbb{L}M$ is a complete lattice with partial order $\sqsubseteq \subseteq \mathbb{L}M \times \mathbb{L}M$ given by $\alpha \sqsubseteq \beta$ if and only if $\alpha(x) \leq \beta(x)$ for all $x \in M$, and with an addition $\oplus$ given by $(\alpha \oplus \beta)(x) = \alpha(x) + \beta(x)$. The bottom element of $\mathbb{L}M$ is also the zero of $\oplus$ and given by $\bot(x) = 0$, and the top element is $\top(x) = \infty$.

Definition 5. *A* recursive specification *of a trace distance* td *consists of*

- *a set* M *with a lattice homomorphism* $\mathsf{eval} : \mathbb{L}M \to \mathbb{R}_{\geq 0} \cup \{\infty\}$,

- *an $\mathbb{L}M$-hemimetric $td^{\mathbb{L}M} : \Sigma^\infty \times \Sigma^\infty \to \mathbb{L}M$ which satisfies $td = \mathsf{eval} \circ td^{\mathbb{L}M}$ and $td^{\mathbb{L}M}(a,b) = \bot$ for all $a, b \in \Sigma$ with $a \sqsubseteq b$, and*
- *a function $F : \Sigma \times \Sigma \times \mathbb{L}M \to \mathbb{L}M$.*

F must be monotone in the third coordinate and satisfy, for all $a, b \in \Sigma$ and $\sigma, \tau \in \Sigma^\infty$, that $td^{\mathbb{L}M}(a.\sigma, b.\tau) = F(a, b, td^{\mathbb{L}M}(\sigma, \tau))$.

Note that the definition implies that for all $a, b \in \Sigma$, $td^{\mathbb{L}M}(a,b) = td^{\mathbb{L}M}(a.\boldsymbol{\varepsilon}, b.\boldsymbol{\varepsilon}) = F(a, b, td^{\mathbb{L}M}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon})) = F(a, b, \bot)$. Hence also $F(a, a, \bot) = td^{\mathbb{L}M}(a,a) = \bot$ for all $a \in \Sigma$.

We have shown in [2, 10, 12] that all commonly used trace distances obey a recursive characterization as above. The point-wise distance from [8], for example, has $\mathbb{L} = \mathbb{R}_{\geq 0} \cup \{\infty\}$, $\mathsf{eval} = \mathsf{id}$ and $d_{\mathsf{m}}^{\mathbb{L}M}(a.\sigma, b.\tau) = \max(d(a,b), d_{\mathsf{m}}^{\mathbb{L}M}(\sigma, \tau))$, where $d : \Sigma \times \Sigma \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is a hemimetric on labels. The limit-average distance used in *e.g.* [7] has $\mathbb{L} = (\mathbb{R}_{\geq 0} \cup \{\infty\})^{\mathbb{N}}$, the complete lattice of functions $\mathbb{N} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$, $\mathsf{eval}(\alpha) = \liminf_{j \in \mathbb{N}} \alpha(j)$ and $d_{\mathsf{m}}^{\mathbb{L}M}(a.\sigma, b.\tau)(j) = \frac{1}{j+1} d(a,b) + \frac{j}{j+1} d_{\mathsf{m}}^{\mathbb{L}M}(\sigma, \tau)$.

For the rest of this section, we fix a recursively specified trace distance. A WMTS $(S, s^0, \dashrightarrow, \longrightarrow)$ is *deterministic* if it holds for all $s \in S$, $s \overset{a_1}{\dashrightarrow} s_1$, $s \overset{a_2}{\dashrightarrow} s_2$ for which there is $a \in \Sigma$ with $td^{\mathbb{L}M}(a, a_1) \neq \top$ and $td^{\mathbb{L}M}(a, a_2) \neq \top$ that $a_1 = a_2$ and $s_1 = s_2$.

Definition 6. *The* lifted modal refinement distance $d_{\mathsf{m}}^{\mathbb{L}M} : S_1 \times S_2 \to \mathbb{L}$ *between the states of WMTS $\mathcal{S}_1 = (S_1, s_1^0, \dashrightarrow_1, \longrightarrow_1)$, $\mathcal{S}_2 = (S_2, s_2^0, \dashrightarrow_2, \longrightarrow_2)$ is defined to be the least fixed point to the equations*

$$d_{\mathsf{m}}^{\mathbb{L}M}(s_1, s_2) = \max \begin{cases} \sup\limits_{s_1 \overset{a_1}{\dashrightarrow}_1 t_1} \inf\limits_{s_2 \overset{a_2}{\dashrightarrow}_2 t_2} F(a_1, a_2, d_{\mathsf{m}}^{\mathbb{L}M}(t_1, t_2)), \\ \sup\limits_{s_2 \overset{a_2}{\longrightarrow}_2 t_2} \inf\limits_{s_1 \overset{a_1}{\longrightarrow}_1 t_1} F(a_1, a_2, d_{\mathsf{m}}^{\mathbb{L}M}(t_1, t_2)). \end{cases}$$

We let $d_{\mathsf{m}}^{\mathbb{L}M}(\mathcal{S}_1, \mathcal{S}_2) = d_{\mathsf{m}}^{\mathbb{L}M}(s_1^0, s_2^0)$. The *modal refinement distance* is $d_{\mathsf{m}} = \mathsf{eval} \circ d_{\mathsf{m}}^{\mathbb{L}M}$, and we write $\mathcal{S}_1 \leq_{\mathsf{m}}^{\varepsilon} \mathcal{S}_2$, for $\varepsilon \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, if $d_{\mathsf{m}}^{\mathbb{L}M}(\mathcal{S}_1, \mathcal{S}_2) \leq \varepsilon$.

Proposition 1. *The modal refinement distance is a well-defined hemimetric, and $\mathcal{S}_1 \leq_{\mathsf{m}} \mathcal{S}_2$ implies $\mathcal{S}_1 \leq_{\mathsf{m}}^0 \mathcal{S}_2$.*

The *thorough refinement distance* between WMTS $\mathcal{S}_1$, $\mathcal{S}_2$ is

$$d_{\mathsf{t}}(\mathcal{S}_1, \mathcal{S}_2) = \sup_{\mathcal{I}_1 \in [\![\mathcal{S}_1]\!]} \inf_{\mathcal{I}_2 \in [\![\mathcal{S}_2]\!]} d_{\mathsf{m}}(\mathcal{I}_1, \mathcal{I}_2),$$

and we write $\mathcal{S}_1 \leq_{\mathsf{t}}^{\varepsilon} \mathcal{S}_2$, for $\varepsilon \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, if $d_{\mathsf{t}}(\mathcal{S}_1, \mathcal{S}_2) \leq \varepsilon$. As for the modal distance, d_{t} is a hemimetric, and $\mathcal{S}_1 \leq_{\mathsf{t}} \mathcal{S}_2$ implies $\mathcal{S}_1 \leq_{\mathsf{t}}^0 \mathcal{S}_2$.

Theorem 1. *For all WMTS $\mathcal{S}_1$, $\mathcal{S}_2$, $d_{\mathsf{t}}(\mathcal{S}_1, \mathcal{S}_2) \leq d_{\mathsf{m}}(\mathcal{S}_1, \mathcal{S}_2)$. If $\mathcal{S}_2$ is deterministic, then $d_{\mathsf{t}}(\mathcal{S}_1, \mathcal{S}_2) = d_{\mathsf{m}}(\mathcal{S}_1, \mathcal{S}_2)$.*

3.3 Conjunction

Let $\owedge : \Sigma \times \Sigma \hookrightarrow \Sigma$ be a commutative partial *label conjunction* operator for which it holds, for all $b_1, b_2 \in \Sigma$, that there is $a \in \Sigma$ for which both $td^{\mathbb{L}M}(a, b_1) \neq \top$ and $td^{\mathbb{L}M}(a, b_2) \neq \top$ iff there exists $c \in \Sigma$ for which both $b_1 \owedge c$ and $b_2 \owedge c$ are defined. This is to relate determinism (left-hand side of the above) to a similar property for label conjunction which is needed in the proof of Theorem 2.

Additionally, we assume that $\owedge$ is greatest lower bound on labels, *i.e.*

- for all $a, b \in \Sigma$ with $a \owedge b$ defined, $a \owedge b \sqsubseteq a$ and $a \owedge b \sqsubseteq b$;
- for all $a, b, c \in \Sigma$ with $a \sqsubseteq b$ and $a \sqsubseteq c$, $b \owedge c$ is defined and $a \sqsubseteq b \owedge c$.

In the definition below, we denote by $\rho_B(\mathcal{S})$ the *pruning* of a WMTS $\mathcal{S} = (S, s^0, \dashrightarrow, \longrightarrow)$ with respect to the states in a ("bad") subset $B \subseteq S$, which is obtained as follows: Define a must-predecessor operator $\mathsf{pre} : 2^S \to 2^S$ by $\mathsf{pre}(S') = \{s \in S \mid \exists a \in \Sigma, s' \in S' : s \xrightarrow{a} s'\}$ and let pre^* be the reflexive, transitive closure of pre. Then $\rho_B(\mathcal{S})$ is defined if $s^0 \notin \mathsf{pre}^*(B)$, and in that case, $\rho_B(\mathcal{S}) = (S_\rho, s^0, \dashrightarrow_\rho, \longrightarrow_\rho)$ with $S_\rho = S \setminus \mathsf{pre}^*(B)$, $\dashrightarrow_\rho = \dashrightarrow \cap (S_\rho \times \Sigma \times S_\rho)$, and $\longrightarrow_\rho = \longrightarrow \cap (S_\rho \times \Sigma \times S_\rho)$.

Definition 7. *The* conjunction *of two WMTS $\mathcal{S}_1 = (S_1, s_1^0, \dashrightarrow_1, \longrightarrow_1)$, $\mathcal{S}_2 = (S_2, s_2^0, \dashrightarrow_2, \longrightarrow_2)$ is the WMTS $\mathcal{S}_1 \wedge \mathcal{S}_2 = \rho_B(S_1 \times S_2, (s_1^0, s_2^0), \dashrightarrow, \longrightarrow)$ given as follows (if it exists):*

$$\frac{s_1 \xrightarrow{a_1}_1 t_1 \quad s_2 \overset{a_2}{\dashrightarrow}_2 t_2 \quad a_1 \owedge a_2 \text{ defined}}{(s_1, s_2) \xrightarrow{a_1 \owedge a_2} (t_1, t_2)} \qquad \frac{s_1 \overset{a_1}{\dashrightarrow}_1 t_1 \quad s_2 \xrightarrow{a_2}_2 t_2 \quad a_1 \owedge a_2 \text{ defined}}{(s_1, s_2) \xrightarrow{a_1 \owedge a_2} (t_1, t_2)}$$

$$\frac{s_1 \overset{a_1}{\dashrightarrow}_1 t_1 \quad s_2 \overset{a_2}{\dashrightarrow}_2 t_2 \quad a_1 \owedge a_2 \text{ defined}}{(s_1, s_2) \overset{a_1 \owedge a_2}{\dashrightarrow} (t_1, t_2)}$$

$$\frac{s_1 \xrightarrow{a_1}_1 t_1 \quad \forall s_2 \overset{a_2}{\dashrightarrow}_2 t_2 : a_1 \owedge a_2 \text{ undef.}}{(s_1, s_2) \in B} \qquad \frac{s_2 \xrightarrow{a_2}_2 t_2 \quad \forall s_1 \overset{a_1}{\dashrightarrow}_1 t_1 : a_1 \owedge a_2 \text{ undef.}}{(s_1, s_2) \in B}$$

Note that conjunction of WMTS may give inconsistent states which need to be pruned away after. As seen in the last two SOS rules above, this is the case when one WMTS specifies a must-transition which the other WMTS cannot synchronize with. Here, the demand on implementations of the conjunction would be that they simultaneously *must* and *cannot* have a transition, which of course is unsatisfiable.

Theorem 2. *Let $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ be WMTS.*

- *If $\mathcal{S}_1 \wedge \mathcal{S}_2$ is defined, then $\mathcal{S}_1 \wedge \mathcal{S}_2 \leq_{\mathsf{m}} \mathcal{S}_1$ and $\mathcal{S}_1 \wedge \mathcal{S}_2 \leq_{\mathsf{m}} \mathcal{S}_2$.*
- *If $\mathcal{S}_1 \leq_{\mathsf{m}} \mathcal{S}_2$, $\mathcal{S}_1 \leq_{\mathsf{m}} \mathcal{S}_3$, and $\mathcal{S}_2$ or $\mathcal{S}_3$ is deterministic, then $\mathcal{S}_2 \wedge \mathcal{S}_3$ is defined and $\mathcal{S}_1 \leq_{\mathsf{m}} \mathcal{S}_2 \wedge \mathcal{S}_2$.*

3.4 Structural composition

Let $\oplus : \Sigma \times \Sigma \hookrightarrow \Sigma$ be a commutative partial *label composition* operator which specifies which labels can synchronize. Again we need to relate determinism to an analogous property for label composition, hence we require that it holds, for all $b_1, b_2 \in \Sigma$, that there is $a \in \Sigma$ for which both $d(a, b_1) \neq \top_{\mathbb{L}}$ and $d(a, b_2) \neq \top_{\mathbb{L}}$ iff there exists $c \in \Sigma$ for which both $b_1 \oplus c$ and $b_2 \oplus c$ are defined.

Additionally, we assume that there exists a function $P : \mathbb{L} \times \mathbb{L} \to \mathbb{L}$ which allows us to infer bounds on distances on synchronized labels. We assume that P is monotone in both coordinates, has $P(\bot_{\mathbb{L}}, \bot_{\mathbb{L}}) = \bot_{\mathbb{L}}$, $P(\alpha, \top_{\mathbb{L}}) = P(\top_{\mathbb{L}}, \alpha) = \top_{\mathbb{L}}$ for all $\alpha \in \mathbb{L}$, and that

$$F(a_1 \oplus a_2, b_1 \oplus b_2, P(\alpha_1, \alpha_2)) \sqsubseteq_{\mathbb{L}} P(F(a_1, b_1, \alpha_1), F(a_2, b_2, \alpha_2)) \tag{1}$$

for all $a_1, b_1, a_2, b_2 \in \Sigma$ and $\alpha_1, \alpha_2 \in \mathbb{L}$ for which $a_1 \oplus a_2$ and $b_1 \oplus b_2$ are defined. Hence $d(a_1 \oplus a_2, b_1 \oplus b_2) \sqsubseteq P(d(a_1, b_1), d(a_2, b_2))$ for all such $a_1, b_1, a_2, b_2 \in \Sigma$.

Intuitively, P gives a *uniform bound* on label composition: distances between composed labels can be bounded above using P and the individual labels' distances, and (1) ensures that this bound holds recursively.

Definition 8. *The* structural composition *of two WMTS* $\mathcal{S}_1 = (S_1, s_1^0, \dashrightarrow_1, \longrightarrow_1)$, $\mathcal{S}_2 = (S_2, s_2^0, \dashrightarrow_2, \longrightarrow_2)$ *is the WMTS* $\mathcal{S}_1 \| \mathcal{S}_2 = (S_1 \times S_2, (s_0^1, s_0^2), \dashrightarrow, \longrightarrow)$ *with transitions defined as follows:*

$$\frac{s_1 \overset{a_1}{\dashrightarrow}_1 t_1 \quad s_2 \overset{a_2}{\dashrightarrow}_2 t_2 \quad a_1 \oplus a_2 \text{ def.}}{(s_1, s_2) \overset{a_1 \oplus a_2}{\dashrightarrow} (t_1, t_2)} \qquad \frac{s_1 \overset{a_1}{\longrightarrow}_1 t_1 \quad s_2 \overset{a_2}{\longrightarrow}_2 t_2 \quad a_1 \oplus a_2 \text{ def.}}{(s_1, s_2) \overset{a_1 \oplus a_2}{\longrightarrow} (t_1, t_2)}$$

The next theorem shows that structural composition supports *quantitative independent implementability*: the distance between structural compositions can bounded above using P and the distances between the individual components.

Theorem 3. *For all WMTS* $\mathcal{S}_1$, $\mathcal{T}_1$, $\mathcal{S}_2$, $\mathcal{T}_2$ *with* $d_{\mathsf{m}}(\mathcal{S}_1 \| \mathcal{S}_2, \mathcal{T}_1 \| \mathcal{T}_2) \neq \top_{\mathbb{L}}$, *we have* $d_{\mathsf{m}}(\mathcal{S}_1 \| \mathcal{S}_2, \mathcal{T}_1 \| \mathcal{T}_2) \sqsubseteq_{\mathbb{L}} P(d_{\mathsf{m}}(\mathcal{S}_1, \mathcal{T}_1), d_{\mathsf{m}}(\mathcal{S}_2, \mathcal{T}_2))$.

References

1. Sebastian S. Bauer, Uli Fahrenberg, Line Juhl, Kim G. Larsen, Axel Legay, and Claus Thrane. Quantitative refinement for weighted modal transition systems. In *MFCS*, volume 6907 of *LNCS*, pages 60–71. Springer, 2011.
2. Sebastian S. Bauer, Uli Fahrenberg, Axel Legay, and Claus Thrane. General quantitative specification theories with modalities. In *CSR*, volume 7353 of *LNCS*, pages 18–30. Springer, 2012.
3. Sebastian S. Bauer, Line Juhl, Kim G. Larsen, Axel Legay, and Jiří Srba. Extending modal transition systems with structured labels. *Mathematical Structures in Computer Science*, 22(4):581–617, 2012.
4. Nikola Beneš, Jan Křetínský, Kim G. Larsen, Mikael H. Møller, and Jiří Srba. Parametric modal transition systems. In *ATVA*, volume 6996 of *LNCS*, pages 275–289. Springer, 2011.

5. Nikola Beneš and Jan Křetínský. Process algebra for modal transition systemses. In *MEMICS*, volume 16 of *OASICS*, pages 9–18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2010.
6. Gérard Boudol and Kim G. Larsen. Graphical versus logical specifications. In *CAAP*, volume 431 of *LNCS*, pages 57–71. Springer, 1990.
7. Pavol Černý, Thomas A. Henzinger, and Arjun Radhakrishna. Simulation distances. *Theor. Comput. Sci.*, 413(1):21–35, 2012.
8. Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Mariëlle Stoelinga. Model checking discounted temporal properties. *Theor. Comput. Sci.*, 345(1):139–170, 2005.
9. Luca de Alfaro, Marco Faella, and Mariëlle Stoelinga. Linear and branching system metrics. *IEEE Trans. Software Eng.*, 35(2):258–273, 2009.
10. Uli Fahrenberg, Axel Legay, and Claus Thrane. The quantitative linear-time–branching-time spectrum. In *FSTTCS*, volume 13 of *LIPIcs*, pages 103–114. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.
11. Uli Fahrenberg, Axel Legay, and Louis-Marie Traonouez. Specification theories for probabilistic and real-time systems. In *From Programs to Systems – The Systems Perspective in Computing*, volume 8415 of *LNCS*. Springer, 2014. In this volume.
12. Uli Fahrenberg, Claus R. Thrane, and Kim G. Larsen. Distances for weighted transition systems: Games and properties. In *QAPL*, volume 57 of *Electr. Proc. Theor. Comput. Sci.*, pages 134–147, 2011.
13. Harald Fecher and Heiko Schmidt. Comparing disjunctive modal transition systems with an one-selecting variant. *J. Logic Alg. Program.*, 77(1-2):20–39, 2008.
14. Patrice Godefroid, Michael Huth, and Radha Jagadeesan. Abstraction-based model checking using modal transition systems. In *CONCUR*, volume 2154 of *LNCS*, pages 426–440. Springer, 2001.
15. Susanne Graf and Joseph Sifakis. A logic for the description of non-deterministic programs and their properties. *Inf. Control*, 68(1-3):254–270, 1986.
16. Alexander Gruler, Martin Leucker, and Kathrin D. Scheidemann. Modeling and model checking software product lines. In *FMOODS*, volume 5051 of *LNCS*, pages 113–131. Springer, 2008.
17. Sören Holmström. A refinement calculus for specifications in Hennessy-Milner logic with recursion. *Formal Asp. Comput.*, 1(3):242–272, 1989.
18. Michael Huth, Radha Jagadeesan, and David A. Schmidt. Modal transition systems: A foundation for three-valued program analysis. In *ESOP*, volume 2028 of *LNCS*, pages 155–169. Springer, 2001.
19. Line Juhl, Kim G. Larsen, and Jiří Srba. Modal transition systems with weight intervals. *J. Log. Algebr. Program.*, 81(4):408–421, 2012.
20. Kim G. Larsen. A context dependent equivalence between processes. *Theor. Comput. Sci.*, 49:184–215, 1987.
21. Kim G. Larsen. Modal specifications. In *Automatic Verification Methods for Finite State Systems*, volume 407 of *LNCS*, pages 232–246. Springer, 1989.
22. Kim G. Larsen, Ulrik Nyman, and Andrzej Wąsowski. On modal refinement and consistency. In *CONCUR*, volume 4703 of *LNCS*, pages 105–119. Springer, 2007.
23. Kim G. Larsen and Bent Thomsen. A modal process logic. In *LICS*, pages 203–210. IEEE Computer Society, 1988.
24. Kim G. Larsen and Liu Xinxin. Equation solving using modal transition systems. In *LICS*, pages 108–117. IEEE Computer Society, 1990.
25. Radu Mardare and Alberto Policriti. A complete axiomatic system for a process-based spatial logic. In *MFCS*, volume 5162 of *LNCS*, pages 491–502. Springer, 2008.

26. Sebastian Nanz, Flemming Nielson, and Hanne Riis Nielson. Modal abstractions of concurrent behaviour. In *SAS*, volume 5079 of *LNCS*, pages 159–173. Springer, 2008.
27. Jean-Baptiste Raclet, Eric Badouel, Albert Benveniste, Benoît Caillaud, and Roberto Passerone. Why are modalities good for interface theories? In *ACSD*, pages 119–127. IEEE, 2009.
28. Sebastián Uchitel and Marsha Chechik. Merging partial behavioural models. In *FSE*, pages 43–52. ACM, 2004.